

### **Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings of claims in the application.

### **Listing of Claims:**

1. (Currently Amended) A method of safeguarding at least one safety-critical program part ~~that is critical to safety~~ against inadvertent execution in a processor, comprising:

providing a first memory sector and a second memory sector in a program memory, wherein the at least one safety-critical program part is contained in the second memory sector, and wherein at least one further program part is contained in the first memory sector, and wherein the first memory sector is configured to be executed prior to the second memory sector;

~~executing the at least one further program part in a predetermined chronological sequence; at a certain point in time in the execution, wherein the at least one further program part includes a routine for~~ generating a pattern using program code ~~that is included in the at least one program part; and~~

at least at one point in time after the certain point in time when the pattern is configured to be generated, checking whether the pattern is present;

wherein a program code for resetting the processor is stored in a section of the program memory corresponding to a chronological execution point prior to the safety-critical program part.

2. (Currently Amended) The method of claim 1, wherein the pattern is generated at a beginning of the execution of the at least one further program part.

3. (Previously Presented) The method of claim 1, wherein the pattern is stored in a volatile memory element.

4. (Original) The method of claim 1, further comprising:

checking an external boundary condition at the time of pattern generation and pattern checking.

5. (Original) The method of claim 4, wherein a state of a hardware component serves as the external boundary condition.

6. (Previously Presented) A method of safeguarding a safety-critical program part ~~that is critical to safety~~ against inadvertent execution in a processor, comprising:

providing a first memory sector and a second memory sector in a program memory, wherein the safety-critical program part is contained in the second memory sector, and wherein at least one further program part is contained in the first memory sector, and wherein the first memory sector is configured to be executed prior to the second memory sector;

~~performing a check at least at one point in time during an execution of the program part that is critical to safety~~ to determine a presence of a pattern representing a proper sequence of ~~the program part execution~~, the pattern being configured to be generated using code that is included in the at least one further program part; and

~~terminating the program execution of the program part~~ if the pattern is determined to be not present, wherein a program code for resetting the processor is stored in a section of the program memory corresponding to a chronological execution point prior to the safety-critical program part.

7. (Withdrawn) A memory device for storing program instructions to cause a microprocessor to safeguard at least one program part that is critical to safety against inadvertent execution, the microprocessor being divided into at least one area, each area storing a respective one of the at least one program part, the at least one program part being executable in a predetermined chronological sequence, the memory device comprising:

a first arrangement for generating a pattern at a certain point in time when the at least one program part is executed; and

at least one second arrangement for performing a check at a later point in time to determine whether the pattern is present.

8. (Withdrawn) The memory device of claim 7, further comprising:

an arrangement for resetting the microprocessor.